

1-- Imperative Programming ve Declarative Programming kavramlarını kısaca açıklayıp farklarını belirtiniz.

Declarative programmingde biz ne istediğimizi belirtiriz/kodlarız. Kontrol akışını direkt olarak belirtmeden yapılır. Imperative programmingde her şey programcı tamamen girilir, kontrol tamamen yazılımcıdadır. Adım adım her şey söylenir. Mesela örnek olarak şunu verebiliriz, bir benzetme olarak; Imperativev restorana gider ve 300gr biftek, az pişmiş, yanında patates ve ketçap ile salata da olsun der. Declarative ise sadece 50 lira ödemek stiyorum ve biftek yemek istiyorum der.

2-- Veri tabanlarının sorgu optimizasyonlarında index oluşturmanın avantajı nedir ? Sık index kullanmak bir probleme yol açar mı?

Indexleme sayesinde veritabanlarında veri düzenlemesi belli bir sıraya göre yapılır. Bu sayede yeni bir sordu oluşturulduğunda, db'de nereye gidileceği en başta bilinir. Bu sayede sorgu sürei ciddi anlamda kısalır. Az veri olan Dbde bu çok büyük bir avantaj olmayabilir ama data büyüdükçe ciddi performans artışı olacaktır. Fakat bu yöntemde sürekli ve her yerde kullanılmamalıdır. Neden? Çünkü düzenleme demek Db üzerinde işlem demek yeniden dataların yazılması vs vs demek. Bu sebeple sürekli data yazılacak olan yerlerde bu problemlere yol açar, sürekli bi düzenleme demek.

3-- İlişkisel veritabanları için normalizasyon kavramı neyi ifade etmektedir ? İlk 3 normal formu örnek üzerinden açıklayınız.

Normalizasyon demek aslında dataların minimum bağımlı değişkenleri ile tek bir tabloda tutulması demektir. Bu sayede aslında tekrardan arındırılmış daha fazla sayıda ama daha az satırlık (genelde) tablolar ile veri saklanır. Mesela bir IMDB örneği düşünelim. Filmlerin database'in MovieID, Movie Name, Country, Language, ActorID gibi fieldleri barındırdığını düşünelim. 1. Normalizasyon aslında hepsinin olduğu şeklidir. 2.Normal formda MovieID, Country, Language ayrılır, çünkü Language Country'e, Country de MovieID'ye bağlıdır. 3. Normal formda ise direkt olarak Country ve sadece Language tutulabilir çünkü Country'den language türer sadece ona bağlıdır.

4-- ORM kütüphaneleri kullanmak her zaman avantajlı mıdır ? ORM kütüphanelerinin ne gibi dezavantajları olabilir ?

ORM sayesinde DB ile kod arasında bir bağımsızlık yaratılabilir. Fakat işler kompleksleşip daha detay işler yapılmaya başlandığında ORM araçlarından direkt olarak faydalanamayıp karışık sorgular için direkt native SQL yazmak zorunda kalabiliriz. Ayrıca ORM kullanmak hızı etkileyen bir yöntemdir. DB ile kod arasına ekstaradan konan bir katman olduğu için performans etkilenir.

5-- Domain Specific Language (DSL) kavramını açıklayınız.

DSL belli bir aplikasyon için özel özel olara oluşturmuş dildir. Farklı domainlerde çalışan diller yerine DSLler belirli problemler için geliştirilmiştir. Bunlara sadece web sayfaları tarafından kullanılan HTML,

sadece belli uygulamaların anladığı keyword olan programların dilleri veya BDDde kullanılan test caseleri yazdığım Gherkin language örnek verilebilir.

6-- Long lived transaction kavramı hangi tip transactionları ifade etmektedir ? Dezavantajları var mıdır ? Varsa nelerdir ?

Long lived transaction kavramı biren fazla veritabanına işleyen transactionları ifade eder. Dezavantajları normal tek DB ile çalışmaktan daha zor olmasıdır. Bu durum eşzamanlılık denetimini ve ölçeklenebilirliği güç kılar.

7-- Thread Pool nedir ? Nerelerde kullanılır ?

Thread pool, sürekli bir iş için yeni thread oluşturmaktan kaçınmak için boşta olan threadlerin alınıp kullanılması mantığı ile işleyen bir süreçte bulunur. Yani işlemde boşta olan ve kullanılmayı bekleyen threadlerin beklediği yer olarak düşünebiliriz. Havuza gelen talepler, havuz içerisinde bir thread'e atanır. dosya giriş çıkışı(IO), yapay zeka, veritabanı, Karmaşık Matematik problemlerin çözüm algoritmaları gibi Multi-Threading gerektiren işlemlerde değerlendirilebilir.

8-- Scalability nedir ? Horizontal ve Vertical Scalability kavramlarını açıklayınız.

Yatay ölçeklendirme , kaynak havuzunuza daha fazla makine ekleyerek ölçeklendirdiğiniz anlamına gelirken, Dikey ölçekleme mevcut bir makineye daha fazla güç (CPU, RAM) ekleyerek ölçeklendirdiğiniz anlamına gelir . Bunu hatırlamanın kolay bir yolu, bir sunucu rafındaki bir makineyi düşünmektir, yatay yönde daha fazla makine ekler ve dikey yönde bir makineye daha fazla kaynak ekleriz .

9-- Data replication ve sharding nedir ? Aralarında nasıl bir fark bulunmaktadır?

Data replication datanın yatay ölçekleme ile aynı datanın farklı nodelara dağıtılmasıdır. Sharding ise datanın ufak parçalara bölünüp farklı nodelarda saklanmasıdır.